基于稠密连接记忆神经网络的文本推理 *

潘永华，闭应洲†，潘怀奇，郑思霞

(广西师范学院 计算机与信息工程学院，南宁 530001)

摘　要：由于传统的端到端记忆神经网络模型特征表示能力不足，无法很好地表示各个记忆之间的联系，导致其在数据集 bAbI 中的位置推理和路径查找问题正确率不高。针对此问题，提出了一种结合稠密连接和多层感知机的记忆神经网络。该模型利用稠密连接与全连接层获取关系特征，增强了模型的特征表示能力。利用 bAbI 数据集对模型进行推理正确率的评估，实验结果表明，与传统的记忆神经网络以及端到端记忆神经网络相比，该模型能有效提升文本推理的正确率。

关键词：记忆神经网络；稠密连接；文本推理；多层感知机；特征表示

中图分类号：TP389.1　　doi: 10.19734/j.issn.1001-3695.2018.10.0794

Text reasoning base on densely connective memory networks

Pan Yonghua, Bi Yingzhou†, Pan Huaiqi, Zheng Sixia

(*School of Computer & Information Engineering*, *Guangxi Teachers Education University*, *Nanning* 530001, *China*)

Abstract: Because the traditional end-to-end memory networks model had insufficient feature representation ability, it could not well represented the relationship between each memory, which leds to the low accuracy of location reasoning and path finding in the bAbI dataset. This paper proposed a new memory networks combining density connectivity and multi-layer perceptron to solve this problem. This model used density connectivity and full connected layer to capture relationship features, which enhanced the capability of feature representation. The proposed model evaluated the accuracy of text reasoning using bAbI dataset. The experimental results show that the model can effectively improve the reasoning accuracy compare with traditional memory network and the existing end-to-end memory network.

Key words: memory networks; density connectivity; text reasoning; multi-layer perceptron; feature representation

0　引言

随着如 Siri、Cortana、QQ 小冰等聊天机器人的出现，聊天机器人正越来越广泛地应用于日常生活中。这类聊天机器人不仅需要能够回答用户使用自然语言提出的单个问题，还需要对已有的自然语言文本进行推理，然后回答用户的问题。因此，文本推理受到越来越多的研究人员关注。

深度学习 (deep learning, DL) [1]已经成为当前机器学习领域中最有潜力的发展方向。使用神经网络模型去解决自然语言处理中的问题逐渐成为主流，并在文本分类[2]、文本蕴涵[3]等方面取得了很多成果。然而传统的神经网络无法很好的解决文本推理问题，这是由于在文本推理时需要记忆文本中的上下文并通过上下文推理得到答案，而传统的神经网络无法很好地解决这类长时记忆问题，所以人们开始通过基于记忆模型的神经网络模型去解决文本推理问题，其中最具代表性的模型之一是记忆神经网络 (memory networks, MemNN)[4]，并由此衍生出端到端记忆神经网络 (end-to-end memory networks, MemN2N)[5]。

但是 MemN2N 也有许多不足，这些不足造成了 MemN2N 无法很好地解决 bAbI 数据集[6]中的位置推理和路径查找问题。本文认为这是由于 MemN2N 模型特征表示能力的不足，无法很好地表示各个记忆之间的联系，导致端到端记忆神经网络在文本推理数据集 bAbI 中记忆间关系极为重要的位置推理和路径查找问题正确率不高。

为解决 MemN2N 的问题，本文提出了稠密连接记忆神经网络(densely connective memory network, DenMemN2N)。其主要贡献包括：a)使用稠密连接加强各跳间的信息流，使模型能够使用前面几跳的信息；b)使用门控机制过滤稠密连接得到的信息流，只保留有效信息；c)使用多层感知机获取关系特征，增强了模型在对象关系上的特征表示能力。

实验结果表明，本文方法相比于 MemNNs、MemN2N 和 GMemN2N 的回答问题的正确率有很大提升。

1　相关工作

1.1　记忆神经网络

Chopra 等人提出了 MemNNs，并将其用于文本推理上。MemNNs 主要由记忆 m、输入组件、泛化组件、输出组件和回复组件组成。MemNNs 文本推理的模型结构如图 1 所示。考虑一个文本问答的场景，提供一段事实（即几个句子），针对这段事实进行提问。下面将通过这个场景对 MemNNs 进行讲解。

在 MemNNs 中输入组件(组件 I)负责将输入转换成内部特征表示，将转换过程定义为 $I(x)$，其中 x 表示 MemNNs 的输入。在文本推理中，x 可以是一段事实，也可以是问题。泛化组件(组件 G)主要负责更新记忆 m。m 是由 i 索引的一个对象数组，m_i 表示第 i 个记忆。在文献[6]中直接将 $I(x)$ 作为

收稿日期：2018-10-22；**修回日期：**2019-01-02　**基金项目：**广西科技开发项目（桂科 AC16380106）

作者简介：潘永华（1990-），男（壮族），广西南宁人，硕士研究生，主要研究方向为自然语言处理、深度学习；闭应洲（1967-），男（通信作者），教授，博士，主要研究方向为智能计算、机器学习(byzhou@163.com)；潘怀奇（1990-），男，硕士研究生，主要研究方向为深度学习、自然语言处理；郑思霞（1996-），女，硕士研究生，主要研究方向为深度学习、自然语言处理.

记忆：

$$m_{H(x)} = I(x) \tag{1}$$

其中：H(x) 为选择函数，即组件 G 仅更新索引为 H(x) 的记忆，而不修改记忆 m 的其他部分。在文本推理问题中，式(1)的 x 通常为已有的事实； H(x) 通常是文本出现的位置。

输出组件(组件 O)负责读取记忆并进行推理，如计算哪些相关记忆能得到好的回复。回复组件(组件 R)则将组件 O 的输出作为输入，最后生成可以阅读答案。在文本问答场景下，组件 O 发现相关的记忆，然后 R 生成答案。

$$o = O(I(x), m) \tag{2}$$

$$a = R(o) \tag{3}$$

其中：此时 o 含有所有与答案有关的记忆的特征内表示；a 表示预测答案。

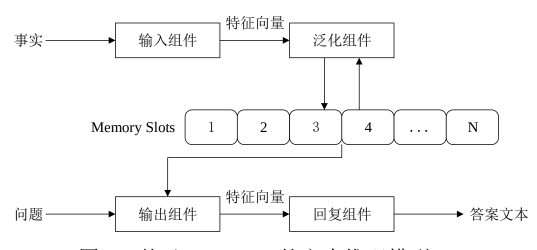


图 1　基于 MemNNs 的文本推理模型

Fig. 1　Text reasoning base on memnns

MemN2N 克服了 MemNNs 无法进行端到端训练的缺点，在数据中进行端到端的学习，学习的目标函数是可微分的函数。受 MemN2N 结构的启发，研究者们提出了许多端到端模型。Kumar 等人[7]针对 MemN2N 只有静态记忆的缺点，提出了一种使用循环神经网络(recurrent neural network, RNN)动态更新记忆的动态记忆神经网络(dynamic memory networks, DMN)。Xiong 等人[8]则对 DMN 的门控循环单元网络 (gated recurrent unit, GRU) [9]进行了改进，提升了计算效率，并且还将提出的模型拓展到了视觉问答（vision question answering, VQA）。门控端到端记忆神经网络(gated end-to-end memory networks, GMemN2N)改进了原 MemN2N 中记忆输入的更新方式，让记忆都要经过门控 T[10]。因此 GMemN2N 的门控机制能够动态地决定记忆如何以及何时跳过推理过程，对记忆网络层的输入进行筛选。本文提出的 DenMemN2N 也使用 GMemN2N 的门控机制筛选记忆。

1.2　增加有效特征

在关于如何获取更多的有效特征的研究中，Srivastava 等人[11]提出了 Highway networks，使用门控方式获取有效特征，解决了网络深度加深、梯度信息回流受阻造成网络训练困难的问题。Huang 等人[12]提出了通过稠密连接，使用前面所有层的输出作为当前层的输入，以此强化层与层之间的信息流。

1.3　关系特征获取

对如何获取两个对象间的关系特征方面，Santoro 等人[13]提出了关系网络(relation network, RN)，并将其用于关系推理上。RN 仅由两个简单的多层感知机（multi-layer perceptron, MLP）组成，其中一个 MLP 结构 $g_\theta(x)$ 对实体之间的关系进行导出，另一个 MLP 结构 $f_\phi(x)$ 使用 $g_\theta(x)$ 获取的实体关系进行推理。由此可看出 MLP 能获取对象间的关系特征。

2　稠密记忆神经网络

2.1　问题描述与定义

在文本推理问题中，通常会有一段事实 X，一个基于这段事实的问题 q 以及答案 a。在本文使用的推理数据中，为了方便评估结果，本文的 a 是一个单词。设 x_i 是一个句子，则事实 X 由一个序列 $x_1, x_2, x_3, \cdots, x_n$ 共 n 个句子组成。设 w_i 代表句子的第 i 个单词，句子的长度是 m。设语料中词汇量的大小为 V。设 f 是一个参数为 θ 的模型，本文期望在数据集中的 a 与预测得到的 a 一致。设数据集中的第 i 个事实 X_i 对应的问题是 q_i， 正确答案是 a_i， 则文本推理可以转换成关于 f 的参数 θ 的参数优化问题，其形式化表达如下：

$$\max_{\theta}(\sum_i \text{equal}(f(X_i, q_i; \theta), a_i)) \tag{4}$$

其中：equal 函数判断模型 f 预测的答案与正确答案是否一致，若一致返回 1，不一致返回 0。

2.2　稠密记忆神经网络

如图 2 所示，稠密记忆神经网络由输入编码模块 I、稠密记忆模块 DM 和输出模块 O 三个主要部分组成。本文将在下面给出各个模块的细节。

2.2.1 输入编码模块

输入编码模块与 MemNNs 中的组件 I 功能类似，主要将模型输入的文本转换成内部特征表示。而在具体实现中，本文使用输入编码模块将每一个句子转换成一个向量表示。它将输入序列的句子 x_i 的每个单词转换成一个向量并求和，用求和得到的向量作为当前句子的向量表示。本文提出的模型将输入的句子中的单词使用独热表示(one-hot representation)进行表示，并使用词嵌入矩阵将单词转换成词向量。设句子 x 中的 w_i 的独热表示为 b_i， 则输入编码模块的形式化表达可表示为

$$e_i = b_i A \tag{5}$$

$$v = \sum_i e_i \tag{6}$$

其中：式(5)的 A 是一个可训练的矩阵，本文称之为嵌入矩阵(embedding matrix)； e_i 是 w_i 的词向量。式(6)将句子 x 中的每个单词的词向量求和得到了句子 x 的向量表示 v，即句子 x 的句向量。

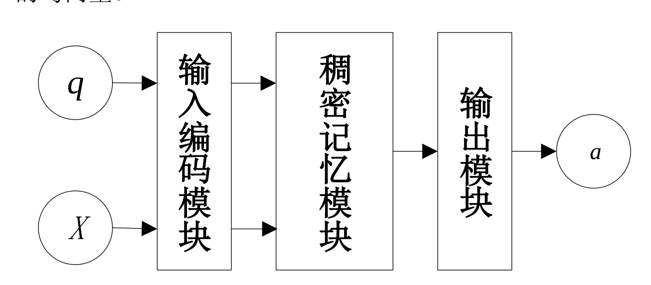


图 2　稠密记忆网络架构

Fig. 2　Densely connective memory networks architecture

2.2.2 稠密记忆模块

稠密记忆模块将 MemNNs 结构的组件 G 和 O 的功能进行了结合，其主要负责对记忆的加工与推理，是实现 DenMemN2N 推理能力的关键。其结构如图 3 所示。

本文更新记忆的方式与 MemNNs 类似，都是将句子的向量表示作为记忆。其中事实 X 与问题 q 的嵌入矩阵都为 A，但事实 X 还使用另一个嵌入矩阵 C 获取 X 中句子的向量表示。其中设 q 的向量化表示为 u_0。 m_i 和 c_i 分别表示 x_i 使用嵌入矩阵 A 与嵌入矩阵 C 向量化的结果， m_i 称为输入记忆， c_i 称为输出记忆。设输入记忆 M_{input} 为 $\{M_{input} | m_1, m_2, \ldots, m_{ms}\}$， 设输出记忆 M_{output} 为 $\{M_{output} | c_1, c_2, \ldots, c_{ms}\}$， ms 表示记忆量(memory size)的大小。

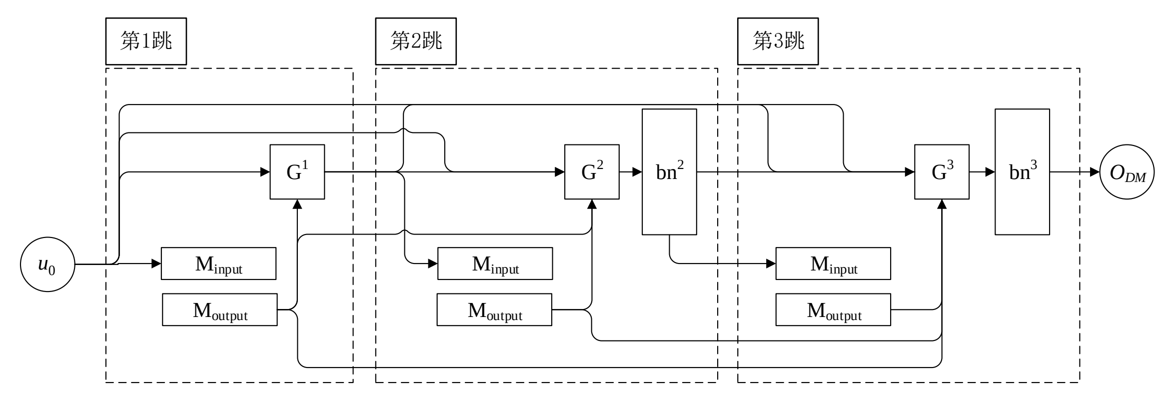


图 3 稠密记忆模块

Fig. 3 Densely memory module

在获取记忆后，DenMemN2N 通过注意力机制，计算问题与各个记忆的相似度，即求两个向量的内积。其计算过程如下：

$$p_i = \mathrm{softmax}((u_0)^T m_i) \tag{7}$$

$$p_i = (u_0)^T m_i \tag{8}$$

式(7)和(8)是获取相似度 p_i 的两种方式。式(7)中的 softmax 函数为 $\mathrm{softmax}(\alpha_i) = e^{\alpha_i} / \sum_j e^{\alpha_j}$ 。在训练过程中，如果训练方式为非线性起始(non-linear start, non-ls)，则通过式(7)获取 p_i；若使用线性起始(linear start, ls)方式，则在训练过程中前 e 代通过式(8)获取 p_i，其中 e 是一个超参。然后将 p_i 与 c_i 的每个元素相乘求和得到 o。

$$o = \sum_i p_i c_i \tag{9}$$

式(9)是实现注意力机制的关键步骤。DenMemN2N 使用注意力机制寻找与问题相关度较高的输入记忆 m_i，越相关 p_i 越高，则 p_i 对应的输出记忆 c_i 的信息就能更多地保留在 o 中，即认为输入的句子与问题相似度越高越有可能与答案相关。

但是若只根据式(9)获取输出，在面对有多个支持事实的问题时，可能无法得到更好的结果。通过表 1 的例子来说明其问题。如果仅通过计算问题与已有事实相似度，在表 1 中第 1 句的相似度是最高的，因为都存在单词“milk”。但是要得出正确答案，无法只根据第 1 句话来得到答案，还需要知道“Mary”最后去到哪里。因此 DenMemN2N 还通过多跳机制进行相关项之间的推理。如图 3 所示，即将第 1 跳得到的输出作为第 2 跳的输入。这种多跳机制实现了自然语言的多级计算，也就是需要将上下文关联的信息在计算中联系起来。让下一跳再通过求相似度的方式找到与上一跳有关的记忆，实质上就是一个联想回忆的过程，使其实现递进式的推理。如表 1 例子中在第 1 跳的输出中第一句的特征相对较多，以此作为第 2 跳的输入。在进行相似度计算时，由于第 1 跳的输出含有“Mary”相关的信息，所以会使同样含有“Mary”的第 4 句权重最高，并可以根据第 4 句获得正确答案。但是现实场景中并非所有推理都是递进式推理，还有通过对象间关系的推理。

表 1 文本推理例子

Table 1 Example of text reasoning

分类	句子	与答案是否相关
已有事实	1 Mary got the milk there.	√
	2 John moved to the bedroom.	
	3 Sandra went back to the kitchen.	
	4 Mary travelled to the hallway.	√
问题与答案	5 Where is the milk? hallway 1 4	

考虑如表 2 中的位置推理问题。答案需要两个事实，并且需要获得形状之间的关系才能获得正确答案。为解决这一问题，本文使用稠密连接增强各跳间的信息流，然后通过门机制筛选出各跳输入和输出中有意义的特征，使模型能更全面地考虑各跳中找到的相关事实，最后将这些相关事实输入到多层感知机中获取关系特征。就如表 2 所示，当第 1 句与第 2 句对物体间的关系进行描述，本文就需要结合两句话的信息，在 DenMemN2N 中，本文使用稠密连接结构增强不同跳中找到的信息并使用门控机制加以过滤，使其过滤一些无用的信息(如例子中的颜色信息)。在获得各跳有效信息后，本文还需要通过这些有效信息获取各个物体（例子中的“triangle”“rectangle”“square”）间的关系特征，最后根据问题与关系特征获取答案。设第 k 跳的门控为 G^k，且 $k \ge 2$。则第 k 跳的计算流程可表示为

$$p_i^k = \mathrm{softmax}((u_i^k)^T m_i) \tag{10}$$

$$o^k = \sum_i p_i^k c_i \tag{11}$$

$$G^k([u^1, u^2, \ldots, u^k]) = \sigma(W_T^k [u^1, u^2, \ldots, u^k] + b_T^k) \tag{12}$$

$$\begin{aligned} u_t^{k+1} = {} & [o^1, o^2, \ldots, o^k] \odot G^k([u^1, u^2, \ldots, u^k]) + \\ & [u^1, u^2, \ldots, u^k] \odot (1 - G^k([u^1, u^2, \ldots, u^k])) \end{aligned} \tag{13}$$

$$u^{k+1} = \mathrm{bn}^k(u_t^{k+1}) \tag{14}$$

其中：式(10)和(11)表示获得第 k 跳的输出 o^k 的流程。式(12)中的 G^k 是一个门函数，$u^1, u^2, \ldots, u^k$ 和 $o^1, o^2, \ldots, o^k$ 分别表示第 1 跳至第 k 跳的记忆层的输入和输出，$\odot$ 表示向量元素对应相乘，σ 表示 Sigmoid 函数。$[u^1, u^2, \ldots, u^k]$ 则表示第 1 跳~第 k 跳输入的串联，$[o^1, o^2, \ldots, o^k]$ 表示第 1 跳至第 k 跳输出的串联。W_T^k 是可以训练的变量，其维度与模型的跳数有关，设句向量的维度是 l_e，则 $W_T^k \in R^{kl_e \times kl_e}$。由于使用了稠密连接，下一跳的输入会得到所有跳的特征，然后所有跳的特征会作为门函数 G^k 的输入并训练参数 W_T^k，其目的是通过该跳前面所有跳的记录去训练 G^k，使得 G^k 能够对 $[u^1, u^2, \ldots, u^k]$ 和 $[o^1, o^2, \ldots, o^k]$ 中的信息进行筛选。式(13)使用 Highway network 的结构对前 k 跳的输入与输出信息进行过滤，最终输出向量 u_t^{k+1}。式(12)与(13)构建了一个使用门 G^k 的稠密连接，增强了跳与跳之间的信息流，使得到的 u_t^{k+1} 不仅包含当前跳的信息，也包含前面几跳的有效信息。

表 2 位置推理例子

Table 2 Example of positional reasoning

	句子	与答案是否相关
事实	1 The triangle is above the pink rectangle.	√
	2 The blue square is to the left of the triangle.	√
问题与答案	3 Is the pink rectangle to the right of the blue square? yes 1 2	

式(14)的 bn^k 表示第 k 跳的批标准化层(batch normalization, bn)层[14]。第 2 跳的 bn 层没有激活函数，而第 3 跳的 bn 层激活函数为 Sigmoid 函数。最后将 bn 层的输出作为下一跳的输入。bn^k 的作用主要是获取 u_t^{k+1} 中的关系特征，u_t^{k+1} 中不仅包含当前跳的信息，也包含前几跳的信息，因此 bn^k 能够同时获取跳与跳之间的关系特征与当前跳的关系特征，最终获得输出 u^{k+1}，并将 u^{k+1} 作为下一跳的输入。

如图 3 所示，第 1 跳的流程中没有设置 bn 层，这是因为第 1 跳前没有前置的层，无须使用稠密连接加强各跳之间的信息流，所以也不需要使用 bn 层获取各跳以及记忆间的关系特征。

稠密记忆模块通过稠密连接增强各跳间记忆的信息流，增加了各跳所找到的关键信息在推理中的作用。并且还使用 MLP 获取关系特征，因此该模块增强了模型的特征表示能力。

2.2.3 输出模块

本模型的输出模块功能主要是将稠密记忆模块的输出作为输入然后输出预测答案 a。输出模块获取答案的公式为

$$\hat{p} = \mathrm{sofxmax}(W_o o_{DM} + b_o) \tag{15}$$

其中：W_o 是可以训练的矩阵；o_{DM} 则是稠密记忆模块的输出。如果预测答案 $\hat{a}$ 仅为一个单词，则 $\hat{p}$ 表示每个单词作为答案的概率分布。

2.3 模型训练

假设问题 q_i 的答案为 a，则输出目标向量的元素值表示为

$$p_i = \begin{cases} 1 & j = a \\ 0 & j \neq a \end{cases} \tag{16}$$

使用交叉熵作为 DenMemN2N 的损失函数，如式(17)所示。

$$L = -\sum_{i=1}^{N} p_i \log(p_i) \tag{17}$$

本文训练的优化方式选用随机梯度下降法。为提升模型的准确率和泛化能力，还进行了如下改进：

a)在输入编码模块使用文献[4]提出的位置编码(position encoding)和时序编码，并随机选择时序编码中的 10%加入噪声。

b)对梯度进行裁剪，若梯度张量的 L2 范数大于 40，则 L2 标准化使用 40 进行标准化，防止梯度爆炸。

c)在预测模块的全连接层以及关系导出的全连接层后加入了 dropout 层，防止过拟合。

d)在更新记忆时，本文使用相邻权重捆绑(adjacent weight typing)，即第 1 跳输入记忆嵌入矩阵为 A_0，输出记忆嵌入矩阵为 C_1，则第 k 跳的输入记忆嵌入矩阵为 $A_k = C_{k-1}$，输出记忆矩阵为 C_k。

3 实验与分析

3.1 数据集与数据预处理

本实验使用的数据集为 bAbI 数据集(使用的 bAbI 数据集版本为 1.2)。表 1 和 2 均为 bAbI 数据集中的某个任务，有一组事实 X、问题 q 以及答案 a，其中事实 X 为一组句子。X 和 q 使用词袋模型表示，a 使用大小为$|V|$的独热向量表示。并且数据集中还标注了与答案有关的事实。本文提出的 DenMemN2N 不使用除答案外的任何监督信息。本文中仅使用 bAbI 的 10 k 数据集进行实验，即每个任务中有 10 000 个问题。

3.2 超参设置

首先将 bAbI 数据集中训练集的 10%作为验证集，用于超参的调优。学习率 η 初始值为 0.01，并且每 25 代减少一半，即变为 $\eta/2$，直到第 100 代才停止降低 η 的值。训练集分批的大小(batch size)为 32，所有的权重和偏置值使用截断正态分布进行随机初始化。其中，用于权重初始化的正态分布期望为 0，标准差为 0.1；用于偏置初始化的值正态分布期望值为 0.5，标准差为 1。模型在训练中只考虑距离问题最近的 50 个句子，将其作为模型的输入数据，所有模型的跳数均为 3。在本文的实验中，本文的词向量维度大小为 20。

为了避免单次训练的误差，本文将每个任务执行 30 次，并选择 30 次测试中得到的最高正确率作为最终的正确率。由于本文使用的是 bAbI 数据集的 1.2 版本，所以本文用新的数据集重新训练了 MemN2N。

3.3 实验结果

模型的训练采用单一训练的方式，即将 bAbI 数据集中每个任务单独进行训练。bAbI 数据集中各个任务的正确率如表 3 所示。

本文将 DenMemN2N 的实验结果与文献[4]的 MemNNs 强监督(strongly supervised, SS)版本以及文献[5]的 MemNNs 弱监督(weakly supervised heuristic, WSH)版本进行了对比。MemNNs 的强监督版本即知道 X 中哪些事实与问题相关，并将这些信息用于训练之中；MemNNs 的弱监督版本则不使用除答案外的任何监督信息进行训练。同时还将实验结果与 MemN2N 和 GMemN2N 进行了对比。MemN2N 以及 GMemN2N 训练过程中仅使用答案作为监督信息。

由表 3 可知，本文提出的 DenMemN2N 模型在任务 17、任务 18 以及任务 19 上都取得了比 MemN2N 以及 GMemN2N 更好的效果。任务 17 与任务 19 的结果比 GMemN2N 分别提升了 25.8%与 2.5%，比 MemN2N 分别提升了 25.4%和 18.5%，比 MemNNs 的 SS 版本分别提升了 22.6%和 30.9%。该实验结果表明，本文方法很好地解决了 MemNNs 和 MemN2N 无法解决的任务 17 和任务 19，克服了传统端到端记忆神经网络的缺点。

本实验也同时对比了 Non-LS 与 LS 两种获取相似度的方法在 MemN2N、GMemN2N 和 DenMemN2N 模型上的差别。通过表 3 的结果可以看出，这两种方法对 path finding 任务和 basic induction 任务有着很大的影响。在使用 LS 时，模型获取了更多的线性特征，因此可以在 basic induction 任务中获得更佳的结果。但是在使用 LS 时，无法获取更多的非线性特征，因此无法在 path finding 上获得更好的结果。而本文提出的 DenMemN2N 模型在 positional reasoning 任务中，使用 Non-LS 的效果要优于 LS。笔者猜测这是因为对象间关系大多是非线性特征，所以使用 Non-LS 时获取了更多的非线性特征，取得了更好的效果。但无论使用哪种方法，本文提出的模型效果在此任务上都要优于 MemN2N 和 GMemN2N。

但是本文提出的模型也有明显的缺陷，任务 16 的正确率都远低于 GMemN2N 以及 MemN2N。其原因是 LS 对任务 16 的影响极大，但 DenMemN2N 的 BN 层的非线性激活函数极大的降低了 LS 的作用。

4 结束语

本文提出了一种使用稠密连接以及门控机制的记忆神经网络——稠密连接记忆神经网络。实验结果表明，本文提出的基于稠密连接、门控机制以及多层感知机的稠密记忆神经网络模型能够有效地解决传统的端到端记忆神经网络模型的特征表示能力不足的问题，更加充分地获取记忆间的关系特征表示，在文本推理问题上得到了更好的结果。但是新模型仍然存在一些可以改进的地方，如网络结构中超参过多、模

型存在过拟合问题以及过参数化问题。因此，在保留现有模型的框架下，构建一个结构更简单、参数更少的模型将会是今后研究的重点。

表 3 bAbI 10k 数据集训练结果

Table 3 Result of bAbI 10k dataset

任务	MemNNs		MemN2N		GMemN2N		DenMemN2N	
	SS	WSH	Non-LS	LS	Non-LS	LS	Non-LS	LS
1:1 supporting fact	**100.0**	99.9	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
2:2 supporting fact	**100.0**	60.4	99.7	99.7	**100.0**	99.9	99.2	**100.0**
3:3 supporting fact	**100.0**	20.5	90.7	92.7	**95.5**	95.0	95.1	93.7
4:2 argument relation	**100.0**	63.4	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
5:3 argument relation	99.8	78.9	99.4	99.4	99.8	**100.0**	**99.9**	99.7
6:yes/no questions	**100.0**	50.1	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
7:counting	96.9	64.9	96.3	97.1	97.3	**98.2**	97.4	97.7
8:lists/sets	99.0	57.3	99.1	99.2	98.6	**99.7**	99.1	99.0
9:simple negation	**100.0**	63.6	99.0	99.2	**100.0**	**100.0**	**100.0**	**100.0**
10:indefinite knowledge	**100.0**	24.0	96.9	97.6	**100.0**	99.8	**100.0**	**100.0**
11:basic coreference	**100.0**	74.7	**100.0**	**100.0**	**100.0**	**100.0**	89.2	**100.0**
12:conjunction	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
13:compound coreference	**100.0**	87.7	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
14:time reasoning	**100.0**	91.3	**100.0**	**100.0**	**100.0**	**100.0**	99.9	**100.0**
15:basic deduction	**100.0**	31.2	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
16:basic induction	**100.0**	49.1	58.6	**100.0**	61.9	**100.0**	46.8	49.9
17:positional reasoning	75.4	48.9	72.6	56.3	70.6	72.2	**98.0**	86.6
18:size reasoning	97.9	54.2	95.3	93.3	97.6	91.5	**99.1**	98.6
19:path finding	68.1	0.0	80.5	33.5	95.8	69.0	**99.0**	68.5
20:agent's motivation	**100.0**	95.6	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**

参考文献：

[1] Lecun Y, Bengio Y, Hinton G. Deep learning [J]. Nature, 2015, 521 (7553): 436-444.

[2] 余本功，张连彬. 基于 CP-CNN 的中文短文本分类研究 [J]. 计算机应用研究, 2018, 35 (4): 1001-1004. (Yu Bengong, Zhang Lianbin. Chinese short text classification based on CP-CNN [J]. Application Research of Computes, 2018, 36 (5): 1001-1004.)

[3] 谭咏梅，刘姝雯，吕学强. 基于 CNN 与双向 LSTM 的中文文本蕴涵识别方法 [J]. 中文信息学报, 2018, 32 (7): 11-19. (Tan Yonmei, Liu Shuwen, Lyu Xueqiang. CNN and BiLSTM based Chinese textual entailment recognition [J]. Journal of Chinses Information Processin, 2018, 32 (7): 11-19.)

[4] Weston J, Chopra S, Bordes A. Memory networks [C]// Proc of the 2nd International Conference on Learning Representations. 2014.

[5] Sukhbaatar S, Weston J, Fergus R. End-to-end memory networks [C]// Proc of the 29th Conference on Neural Information Processing Systems. Montreal: Curran Associates Inc , 2015: 2440-2448.

[6] Weston J, Bordes A, Chopra S, *et al.* Towards ai-complete question answering: a set of prerequisite toy tasks [C]// Proc of the 4th International Conference on Learning Representations. 2016.

[7] Kumar A, Irsoy O, Ondruska P, *et al.* Ask me anything: dynamic memory networks for natural language processing [C]// Proc of the 33rd International Conference on Machine Learning. New York: PMLR, 2016: 1378-1387.

[8] Xiong Caiming, Merity S, Socher R. Dynamic memory networks for visual and textual question answering [C]// Proc of the 33rd International Conference on Machine Learning. New York: PMLR, 2016: 2397-2406.

[9] Cho K, Merrienboer B, Gulcehre C, *et al.* Learning phrase representations using RNN encoder-decoder for Statistical machine translation [C]// Proc of Conference on Empirical Methods in Natural Language Processing. Doha: Association for Computational Linguistics, 2014: 1724-1734.

[10] Liu Fei, Perez J. Gated end-to-end memory networks [C]// Proc of the 15th Conference on European Chapter of the Association for Computational Linguistics: volume 1, Long Papers. Valencia: Association for Computational Linguistics, 2017: 1-10.

[11] Srivastava R K, Greff K, Schmidhuber J. Training very deep networks [C]// Proc of the 29th Neural Information Processing Systems. Montreal: Curran Associates Inc , 2015: 2377-2385.

[12] Huang Gao, Liu Zhuang, Van Der Maaten L, *et al.* Densely connected convolutional networks [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE Press, 2017: 2261-2269.

[13] Santoro A, Raposo D, Barrett D G, *et al.* A simple neural network module for relational reasoning [C]// Proc of the 31st Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc , 2017: 4967-4976.

[14] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift [C]// Proc of the 32nd International Conference on Machine Learning. Lille: PMLR, 2015: 448-456.